

A Knowledge-based Decision Support System for Payload Scheduling

Rajesh Tyagi
Fan T. Tseng
Department of MIS/MSC
University of Alabama in Huntsville
Huntsville, AL 35899
(205) 895-6510

ABSTRACT

This paper presents the development of a prototype Knowledge based Decision Support System, currently under development, for scheduling payloads/experiments on space station missions. The DSS is being built on Symbolics, a lisp machine, using KEE, a commercial knowledge engineering tool.

INTRODUCTION

The task of payload scheduling is rather complicated. It not only involves using algorithms to generate a schedule, it is also very dynamic in nature since the schedule frequently needs to be revised, often at short notices, due to unpredictable and uncontrollable features like equipment failure, power failure, and other emergencies involving a mission.

The traditional approach taken in solving such problems is to develop a decision support system using conventional programming tools. The purpose of this research is to use artificial intelligence/expert system techniques, both hardware and software, to develop a knowledge-based decision support system for this space station scheduling problem.

The DSS should be able to generate a payload/experiments schedule based on the needs and the objectives of the user. It should also be able to update/modify the schedule as those needs and objectives change over the course of the mission.

THE DSS

The knowledge-based DSS is comprised of three components as shown in Figure 1. These components are: a knowledge base, a models base, and a user interface. The knowledge base possesses information on various attributes of the experiments (like power and experiment run time), and the available resources (like power supply). The models base contains analytical and heuristic models that may be used to develop the experiment schedule. And the user interface provides the dialog between the user and the knowledge and models base.

The Knowledge Base

The knowledge base is organized in the form of frames. Each experiment is represented by a frame, the slots of the frame representing different attributes of the experiment. Figure 2 shows the frame corresponding to experiment 'Crystal Growth'. The experiment, which is sponsored by NASA, is to be run only once, the run requiring a power supply of 500 kilowatts over 50 hours, the duration of the experiment. This data is inputted by the user before the schedule is generated. The starting and ending times for the experiment are determined by the analytical model used to generate the schedules, and are automatically placed in their respective slots. The resources are also similarly represented by frames in the

knowledge base, one frame for each resource. Figure 3 illustrates the overall structure of the knowledge base.

FIGURE 1.

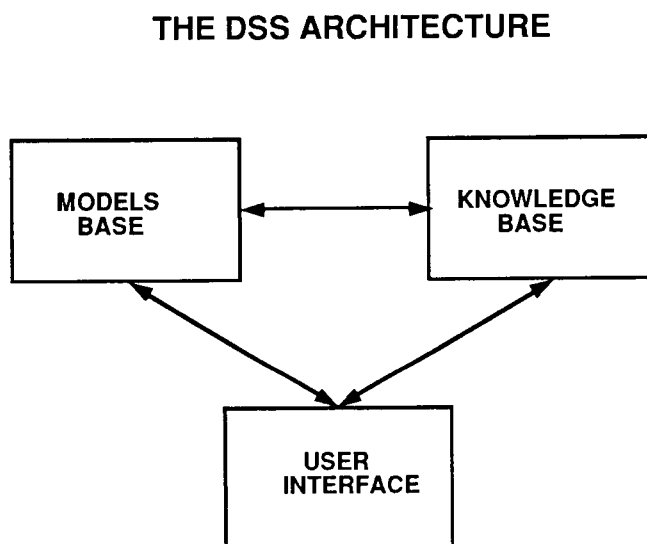


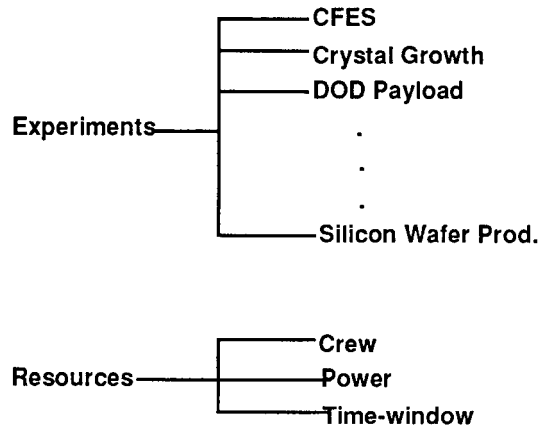
FIGURE 2

Frame for Crystal Growth

<u>SLOT</u>	<u>VALUE</u>
Agency	NASA
Duration	20 hrs
Power	1200 kw
Runs	1
Starting Time	41
Ending Time	60

FIGURE 3.

THE KNOWLEDGE BASE



The Models Base

The models base contains a set of analytical models that may be used to determine a schedule, based on the objectives and/or requirements of the user. The user may specify the criteria that is to be used for scheduling. These criteria include: (1) minimizing the total time needed to schedule a given set of experiments, and (2) leveling the power consumed given that a set of experiments are to be scheduled within a specified amount of time.

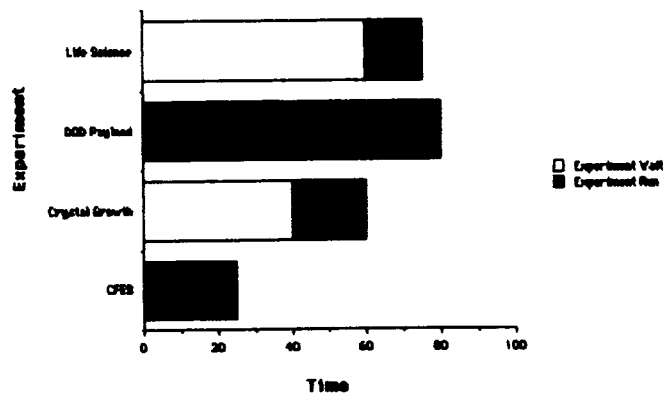
As mentioned earlier, the space station scheduling problem is very complicated, given the set of constraints imposed by the experiments. Most of the current literature deals with developing heuristics to generate the schedule, since the schedules need to be generated in relatively short amount of time, often at short notices, which makes it impractical to use optimizing algorithms which often require a lot of execution time to generate a solution. However, this may not always be a good approach to take since time needed to generate a schedule is not always of the highest concern. In fact, the scheduling task may be broken down into two different phases: (1) constructing an initial schedule, and (2) dynamically modifying the schedule during the mission as needed due to equipment failures etc. The initial schedule, whose planning horizon is spread over a period of weeks/months, could be generated using optimizing algorithms since execution time is not a major concern during this phase. The disadvantage of longer processing times of optimizing algorithms is outweighed by the payoff from a far better schedule that may be generated. It is only during the second phase, when schedules must be modified expediently, that heuristics may be preferable to optimizing algorithms. Thus it is desirable to have a full spectrum of scheduling models ranging from highly optimizing algorithms to 'quick and dirty' heuristics.

The User Interface

The user interface provides the dialog between the user and the DSS in the form of windows, menus, and graphical displays. The user controls the execution of the system by specifying the criteria to be used in determining the schedule, and using menus to update the knowledge base, and updating/modifying the schedule. The output of the DSS is in the form of updated knowledge base and/or graphical displays of the schedule generated. Figure 4 shows an example schedule.

FIGURE 4

EXAMPLE SCHEDULE



CONCLUSION

This paper has presented an overview of a knowledge-based DSS, currently under development, for experiment scheduling on space station missions. The DSS utilizes artificial intelligence/expert system techniques to build the knowledge base and the user interface.

REFERENCES

1. Baker, K. R., Introduction to Sequencing and Scheduling, New York: John Wiley and Sons, Inc. 1974.
2. KEE User's Manual, Intellicorp Inc.
3. Lawler, E. L., Lenstra J. K., and Kan, A. H. G. Rinooy, "Recent Developments in Deterministic Sequencing and Scheduling: A Survey," in M. A. H. Dempster et. al. (Eds.), Deterministic and Stochastic Scheduling, Reidel, Dordrecht, 1982, 35-72.
4. Symbolics software. Report, Symbolics, Inc., 21150 Califa Street, Woodland Hills, California, 1981.
5. Waterman, Donald A., A guide to Expert Systems, Reading, Massachussets: Addison Wesley Publishing Company.
6. Winston, Patrick H., Artificial Intelligence, (2nd ed.), Reading, Massachussets: Addison Wesley Publishing Company, 1984.